Noname manuscript No.
(will be inserted by the editor)

A Quadratic Regularization Method with Finite-Difference Gradient Approximations for Multiobjective Optimization

M.R. Barbosa · J.O. Lopes · P.S.M. Santos · R.V.G. Serra · G.N. Silva

Received: date / Accepted: date

Abstract This paper investigates a quadratic regularization method for multiobjective optimization problems where gradient information is not directly available and must be approximated via finite-difference schemes. We propose an approach that combines a quadratic model with finite-difference gradient approximations to efficiently handle expensive-to-evaluate multiobjective optimization problems. It is shown that the method based on forward finite-difference gradients requires at most $\mathcal{O}(mn\epsilon^{-2})$ function evaluations to achieve an ε-approximate first-order critical point. Numerical experiments comparing the proposed method with the standard steepest descent method for multiobjective optimization demonstrate that the proposed approach requires fewer iterations on average while achieving comparable or improved computational efficiency.

Matheus Rocha Barbosa
CCN, DM, Federal University of Piauí, Teresina, PI , Brazil
E-mail: mathrbarbosa@gmail.com

Jurandir de Oliveira Lopes
CCN, DM, Federal University of Piauí, Teresina, PI , Brazil
E-mail: jurandir@ufpi.edu.br

Paulo Sèrgio Marques dos Santos
Federal University of Delta do Parnaìba, Parnaìba, PI , Brazil
E-mail: paulo.santos@ufdpar.edu.br

Ray Victor Guimarães Serra
CCN, DM, Federal University of Piauí, Teresina, PI , Brazil
E-mail: paulo.santos@ufdpar.edu.br

Gilson do Nascimento Silva
CCN, DM, Federal University of Piauí, Teresina, PI , Brazil
E-mail: gilson.silva@ufpi.edu.br

1 Introduction

In the classical optimization problem that consists of minimizing a continuously differentiable function $f : \mathbb{R}^n \to \mathbb{R}$, the theoretical stopping condition is typically given by the first-order optimality condition, which requires the gradient of f at x to be equal to zero. Nevertheless, in practice, this is commonly replaced by the relaxed criterion $\|\nabla f(x)\| \leq \varepsilon$. Within this practical framework, derivative-free optimization methods gain importance, especially when analyzing their worst-case evaluation complexity. A detailed analysis for the scalar case is available in references such as [8,9,14,13].

In the present study, we analyze the worst-case complexity of a quadratic regularization method (QRM) with finite-difference gradient approximations for multiobjective optimization. The motivation for studying such method with finite-difference comes from the fact, the gradient of some functions involved in practical optimization problems is not readily available or it is expensive to compute, as is the case in aerodynamic shape optimization [12], model calibration [18], tuning of algorithmic parameters [1], and optimization of cardiovascular geometries [16,19].

Recently, there has been a growing interest in the extension and analysis of classical optimization algorithms for solving multiobjective optimization problems:

$$\min_{x \in \mathbb{R}^n} F(x), \tag{1}$$

where $F : \mathbb{R}^n \to \mathbb{R}^m$ is a vector-valued function and the partial order is given by the cone $\mathbb{R}^m_+$. In this setting, a point x^* is a called a weakly Pareto or weakly efficient solution for problem (1) if there is no $x \in \mathbb{R}^n$ such that $f_i(x) < f_i(x^*)$ for all $i \in \mathcal{I} := \{1, \ldots, m\}$, where f_i is the i-th scalar component of F.

Many methods have been proposed for solving problem (1). Most of them are inspired by the well-known methods for solving *scalar* optimization problems, i.e., problem (1) with $m = 1$ and the order given by the cone $\mathcal{K} = \mathbb{R}_+$. In this particular case, QRM is a globally convergent variant of Newton's method for the unconstrained minimization of continuously differentiable functions [2, 4,7,8]. The classical steepest descent method was proposed in [6] for solving continuously differentiable multiobjective optimization problems, i.e., each scalar component of F is continuously differentiable on $\mathbb{R}^n$. This method was extended to the vectorial setting whose the partial order is given by a pointed, convex and closed cone $\mathcal{K} \subset \mathbb{R}^m$ in [5]. Basically, it consists of generating a sequence $\{x^k\}$ given by $x^{k+1} = x^k + t_k v^k$, where $v^k \in \mathbb{R}^n$ is a direction computed as

$$v^k := \arg\min_{v \in \mathbb{R}^n} \max_{i \in \mathcal{I}} \left\{ \langle \nabla f_i(x^k), v \rangle + \frac{1}{2}\|v\|^2 \right\}, \tag{2}$$

or alternatively,

$$v^k := \arg\min_{v \in \mathbb{R}^n} \max_{i \in \mathcal{I}} \left\{ \big(J_F(x^k)v\big)_i + \frac{1}{2}\|v\|^2 \right\}, \tag{3}$$

where J_F denotes the Jacobian matrix of F, and the stepsize t_k is chosen by an Armijo type rule. It is immediate to see that (2)-(3) constitute a strongly convex optimization problem, and hence v^k is uniquely determined. Note that in the scalar case the direction v^k corresponds to the steepest descent direction $-\nabla f(x^k)$ and the classical stopping criterion in the multiobjective setting becomes verifying whether $v(x)$ is equal to zero. Similarly to the scalar case, this scheme with an appropriate line-search generates a decreasing sequence in the partial order given by cone $\mathbb{R}^m_+$, i.e., $f_i(x^{k+1}) \leq f_i(x^k)$, and its convergence properties are similar to its scalar version. In particular, it was shown in [5, 6] that, under some basic assumptions, the whole sequence generated by this scheme converges to a weakly efficient solution for problem (1).

A recent publication on multiobjective optimization is [3], which addresses a high-order regularization method for finding weakly stationary points of problem (1) with constraints under the assumption that the derivatives of the objective functions are Hölder continuous, and hence derivatives of order $p \geq 1$ need to be calculated. Moreover, the algorithm proposed in [3] enforces the sufficient decrease property in the objective function.

Initially, the present work proposes an alternative way to compute (2)-(3) using a derivative-free method. More specifically, we propose approximating the Jacobian matrix J_F with an appropriate finite-difference matrix. The concept of derivative-free gradient methods was recently analyzed, for example, in [7,8]. Thus, we propose to develop versions of [6] using finite-difference based on [7], which proposed and analyzed finite-difference quadratic regularization methods (scalar case). The main goal is based on the fact that first-order multiobjective methods using the Jacobian matrix can be computationally expensive in some applications.

This work is organized as follows. Section 2 presents some preliminaries and notations. In Section 3, we present important auxiliary results used throughout the work. In Section 4, we introduce the proposed method based on a finite-difference approach and state its main result. Section 5 is devoted to the presentation of the numerical experiments.

2 Preliminaries and Notations

In this section, we present some notations and definitions that will be used throughout the paper.

Given a point $q \in \mathbb{R}^n$ and a closed convex set $\mathcal{C} \subset \mathbb{R}^n$, we denote by $P_{\mathcal{C}}q$ the orthogonal projection of q onto $\mathcal{C}$. Let $\mathcal{I} := \{1, \ldots, m\}$ be an index set, and let $A \in \mathbb{R}^{m \times n}$ be a real matrix. We denote by A_i the i-th row of A, for all $i \in \mathcal{I}$. The matrix norm adopted here is defined as the map $\|\cdot\|_{2,\infty} : \mathbb{R}^{m \times n} \to \mathbb{R}$ given by

$$\|A\|_{2,\infty} := \max_{i \in \mathcal{I}} \|A_i\|. \tag{4}$$

It is straightforward to verify that $\|\cdot\|_{2,\infty}$ is indeed a norm.

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a real-valued function. For a given step size $h > 0$, the symbol $d_h f(x)$ denotes a finite-difference approximation of the gradient of

f at a point $x \in \mathbb{R}^n$. Now, if $F : \mathbb{R}^n \to \mathbb{R}^m$ is a vector-valued function, we denote by $JF(x) \in \mathbb{R}^{m\times n}$ the Jacobian matrix of F at point x, defined as

$$JF(x) = \begin{bmatrix} \nabla f_1(x)^T \\ \vdots \\ \nabla f_m(x)^T \end{bmatrix}.$$

and by $D_h F(x) \in \mathbb{R}^{m\times n}$ its finite-difference Jacobian approximation with step size $h > 0$.

Let $A \in \mathbb{R}^{m\times n}$ be a matrix. We denote by $\text{conv}\{A_1, \ldots, A_m\}$ the convex hull of its rows $\{A_1, \ldots, A_m\} \subset \mathbb{R}^n$. In other words, the convex hull operator can be defined as

$$\text{conv}\{A_1, \ldots, A_m\} = \left\{ \sum_{i=1}^{m} \lambda_i A_i \;\middle|\; \lambda \in \Delta_m \right\},$$

where $\Delta_m \subset \mathbb{R}^m$ represents the standard m-simplex, defined as

$$\Delta_m = \left\{ \lambda \in \mathbb{R}^m \;\middle|\; \lambda_i \geq 0 \; \forall i \in \mathcal{I}, \; \sum_{i=1}^{m} \lambda_i = 1 \right\}.$$

Now we define the matrix-to-vector operator $\measuredangle$, which enables us to describe the minimizer in geometric terms, as the projection of the origin onto the convex hull of the rows of A. This leads to an equivalent formulation to a quadratic programming problem, providing a computationally efficient approach to obtaining the steepest descent direction. The matrix-to-vector operator $\measuredangle : \mathbb{R}^{m\times n} \to \mathbb{R}^n$ is defined as

$$\measuredangle A := P_{\mathcal{C}}\, 0; \quad \mathcal{C} := \text{conv}\{A_1, \ldots, A_m\}. \tag{5}$$

In vector and matrix operations, vectors are treated as column vectors by default, but gradient vectors and rows of matrices are treated as row vectors. This is explored to express dot products compactly. For example, expression $A_i v$, where $A \in \mathbb{R}^{m\times n}$, $v \in \mathbb{R}^n$ and A_i is the i-th row of matrix A, results in the dot product between A_i and v, and is used in this work instead of notation $\langle A_i, v\rangle$. Expression $\nabla f v$, where $f : \mathbb{R}^n \to \mathbb{R}$ is a real function and $v \in \mathbb{R}^n$, is also a dot product, since by convention ∇f is a row vector and v is a column vector.

Let $\psi : \mathbb{R}^{m\times n} \times \mathbb{R}^n \to \mathbb{R}$ with

$$\psi(A, v) := \max_i A_i v + \frac{1}{2}\|v\|^2, \tag{6}$$

where A_i is the i-th row of A (and thus $A_i v$ is the dot product between A_i and v).

We recall that the multiobjective gradient method [6] seeks Pareto-critical points by iteratively updating the current point x in the steepest descent direction, which is defined by

$$v(x) := \arg\min_{u \in \mathbb{R}^n} \psi(JF(x), u).$$

The step length of the update is specified by an Armijo-like rule. It is shown in [6] that $v(x) = 0$ if and only if x is Pareto-critical. It is also shown that, if the sequence (x_k) produced by the algorithm has some accumation point, then such accumulation point is Pareto-critical. In a certain sense, this result still holds when the subproblem above is solved inexactly.

3 Auxiliary results

In this paper, the problems under consideration are characterized by the following assumptions regarding F.

A1. *The gradient of f_i is L_i-Lipschitz continuous, i.e.,*

$$\|\nabla f_i(y) - \nabla f_i(x)\| \leq L_i \|y - x\|, \quad \forall x, y \in \mathbb{R}^n, \ \forall i \in \mathcal{I}.$$

For convenience, we denote $L := \max_i L_i$.

A2. *There exists $l_i \in \mathbb{R}$ such that $l_i \leq F_i(x)$ for some $i \in \mathcal{I}$ and for all $x \in \mathbb{R}^n$.*

The analysis in this section focuses on characterizing and approximating the steepest descent direction associated with the mapping $u \mapsto \psi(A, u)$ defined in (6). This direction plays a central role in multiobjective optimization methods, and understanding its properties is essential for designing practical algorithms.

We begin by presenting a result that gives an explicit expression for the minimum value of $\psi(A, u)$ in terms of its minimizer. This result will be important for designing an inexact update direction.

Lemma 1 *Let $A \in \mathbb{R}^{m \times n}$ and*

$$v := \arg\min_{u \in \mathbb{R}^n} \psi(A, u),$$

Then, the minimum value $\alpha := \psi(A, v)$ satisfies $\alpha = -\|v\|^2/2$.

Proof The proof of this result proceeds as in [20], with the Jacobian matrix replaced by a general $m \times n$ matrix A.

The next result gives an alternative characterization for the minimizer of the mapping $u \mapsto \psi(A, u)$.

Lemma 2 *Let $A \in \mathbb{R}^{m\times n}$ and*

$$v := \arg\min_{u} \psi(A, u),$$

where $\psi(A, u)$ is defined in (6). Then, $v = -\curlywedge A$. Equivalently, $v = -A^T\lambda$, where

$$\lambda = \arg\min_{y\in\Delta_m} \|A^T y\|. \tag{7}$$

Proof See Proposition 2.2 and Colorary 2.3 in [20].

Somme comments about Lemma 2 are in order. First, if $A = JF(x)$, Lemma 2 states that the steepest descent direction $v(x)$ of the multiobjective gradient method is $-\curlywedge JF(x)$, the vector opposite to the orthogonal projection of the origin on the convex hull of the gradients $\nabla f_i(x)$, $i \in \mathcal{I}$. Second, it provides a convenient way of computing $v(x)$: (i) initiate by computing λ as in (7); (ii) then, get $v(x) = -A^T\lambda$. This approach is convenient because (7) is a quadratic programming (QP) problem (minimization of a quadratic function under linear constraints), and thus can be solved by efficient QP solvers, widely available. It also makes the opposite vector to the approximation of $v(x)$ be in the convex hull of the gradients $\nabla f_i(x)$, $i \in \mathcal{I}$, which will be useful in this work. Third, in addition to QP solvers, other methods in the literature can be used to solve problem (7), such as the ones in [10] and [15]. Finally, since the multiobjective steepest direction is $-\curlywedge JF(x)$, we can see $\curlywedge JF(x)$ as a generalization of the gradient for vector-valued functions.

The next result establishes a simple yet useful bound on the norm of the operator output $\curlywedge A$. It shows that the projection of the origin onto the convex hull of the matrix rows cannot exceed the smallest norm among those rows.

Lemma 3 *Let $A \in \mathbb{R}^{m\times n}$. Then we have*

$$\|\curlywedge A\| \leq \min_{i\in\mathcal{I}} \|A_i\|.$$

Proof Let $w := \curlywedge A$. Then, it follows from the definition of $\curlywedge$ in (5) that,

$$\|w\| \leq \|z\|, \quad \forall z \in \operatorname{conv}\{A_1, \ldots, A_m\}.$$

Since $A_i \in \operatorname{conv}\{A_1, \ldots, A_m\}$ for all $i \in \mathcal{I}$, we conclude that

$$\|w\| \leq \|A_i\|, \quad \forall i \in \mathcal{I}.$$

Hence, the conclusion now follows from the previous inequality. □

The following lemma is a generalization for the triangle inequality for the operator $\curlywedge$. Its proof is based on the proof of Theorem 3.1 from [20].

Lemma 4 *For any $A, B \in \mathbb{R}^{m\times n}$ there holds:*

$$\|\curlywedge A\| \leq \|\curlywedge B\| + \|A - B\|_{2,\infty}. \tag{8}$$

Proof Let $v_A := \arg\min_u \psi(A,u)$ and $v_B := \arg\min_u \psi(B,u)$, with α_A and α_B being the respective minimum values. Take $j_B \in \arg\max_i B_i u$ and using (6), the fact that $\max_i A_i u \geq A_{j_B} u$, and (4), we have

$$\begin{aligned}\psi(B,u) - \psi(A,u) &\overset{(6)}{=} \left(\max_i B_i u + \frac{1}{2}\|u\|^2\right) - \left(\max_i A_i u + \frac{1}{2}\|u\|^2\right) \\ &= \max_i B_i u - \max_i A_i u = B_{j_B} u - \max_i A_i u \\ &\leq B_{j_B} u - A_{j_B} u \leq \|B_{j_B} - A_{j_B}\|\|u\| \\ &\leq \max_i \|A_i - B_i\|\|u\| \overset{(4)}{=} \|A - B\|_{2,\infty}\|u\|. \end{aligned} \tag{9}$$

As $\psi(B,u)$ is the sum of $\max_i B_i u$, which is convex in u, with $(1/2)\|u\|^2$, in view of (6), we can conclude that $\psi(B,u)$ is 1-strongly convex with respect to u. Hence,

$$\psi(B,u) \geq \alpha_B + \frac{1}{2}\|u - v_B\|^2. \tag{10}$$

Using (9) and (10), we obtain that

$$\|A - B\|_{2,\infty}\|u\| \overset{(9)}{\geq} \psi(B,u) - \psi(A,u) \overset{(10)}{\geq} \alpha_B + \frac{1}{2}\|u - v_B\|^2 - \psi(A,u). \tag{11}$$

Replacing u with v_A in (11) and using the facts that, by Lemma 1, $\alpha_A = -(1/2)\|v_A\|^2$ and $\alpha_B = -(1/2)\|v_B\|^2$, we have

$$\begin{aligned}\|A - B\|_{2,\infty}\|v_A\| &\overset{(11)}{\geq} -\frac{1}{2}\|v_B\|^2 + \frac{1}{2}\|v_A - v_B\|^2 + \frac{1}{2}\|v_A\|^2 \\ &= -\frac{1}{2}\|v_B\|^2 + \frac{1}{2}(\|v_A\|^2 - 2v_A^T v_B + \|v_B\|^2) + \frac{1}{2}\|v_A\|^2 \\ &= \|v_A\|^2 - v_A^T v_B \geq \|v_A\|^2 - \|v_A\|\|v_B\|, \end{aligned}$$

where the last inequality above is due to the Cauchy-Schwarz inequality. Dividing both sides of the previous inequality by $\|v_A\| > 0$ (the case $\|v_A\| = 0$ is immediate), we have

$$\|v_A\| \leq \|v_B\| + \|A - B\|_{2,\infty}.$$

Finally, by Lemma 2, $v_A = -\measuredangle A$ and $v_B = -\measuredangle B$, and thus the last inequality is equivalent to (8), which concludes the proof. □

A natural choice for an inexact solution of problem $\arg\min_u \psi(A,u)$ is $\hat{v} \in \mathbb{R}^n$ such that

$$\psi(A,\hat{v}) \leq \theta \min_{u \in \mathbb{R}^n} \psi(A,u) = -\frac{\theta}{2}\|v_A\|^2, \tag{12}$$

where $\theta \in (0,1]$ and v_A is the exact solution, i.e., $v_A := \arg\min_u \psi(A,u)$. However, unless the exact solution v_A is already known (which will not be the case if we are seeking an inexact solution), (12) can not be directly tested

when computing an inexact solution for the problem in question. The following lemma provides a choice for an inexact solution whose defining relation can be tested without knowing the norm of the exact solution.

Lemma 5 *Let* $A \in \mathbb{R}^{m\times n}$ *and*

$$v_A := \arg\min_u \psi(A,u) \quad \textit{and} \quad \mathcal{A} := \operatorname{conv}\{A_1,\ldots,A_m\}. \tag{13}$$

Let $\theta \in (0,1]$ *and*

$$\mathcal{S} := \left\{ v^+ \in \mathbb{R}^n : v^+ \in \mathcal{A}, \quad \psi(A,v^+) \le -\frac{\theta}{2}\|v^+\|^2 \right\}. \tag{14}$$

Then, the following statements hold:

(a) $\mathcal{S}$ *is non-empty. Moreover, if* $\theta \in (0,1)$*, then* $\mathcal{S}$ *has the same covering dimension as* $\mathcal{A}$*.*

(b) If $v^+ \in \mathcal{S}$*, then*

$$\psi(A,v^+) \le -\frac{\theta}{2}\|v_A\|^2.$$

(c) For $\theta = 1$*, the only* $v^+ \in \mathcal{S}$ *is* $v^+ = v_A$*.*

(d) For $\theta < 1$*, if* v_A *(defined in* (13)*) is an accumulation point of some sequence* $(v_k) \subset \mathcal{A}$*, then there exists* $N \in \mathbb{N}$ *such that* $v_N \in \mathcal{S}$*.*

Proof (a) We first prove that $\mathcal{S}$ is non-empty. It follows from Lemma 2 , (5) and the definition of $\mathcal{A}$ as in (13), that $-v_A = \nwarrow\!A \in \mathcal{A}$, for any $A \in \mathbb{R}^{m\times n}$. Also, Lemma 1 and the assumption on θ imply that $\psi(A,-v_A) = -(1/2)\|v_A\|^2 \le -(\theta/2)\|v_A\|^2$. These two conclusions guarantee that $-v_A \in \mathcal{S}$, and hence that $\mathcal{S}$ is non-empty (the existence of v_A is assured by the fact that v_A is the minimizer of a strictly convex function). We now show the second statement in (a). Assume that $\theta \in (0,1)$ and let $\zeta : \mathbb{R}^n \to \mathbb{R}$ de defined as

$$\zeta(u) := \psi(A,u) + \frac{\theta}{2}\|u\|^2.$$

First observe that the relation $\zeta(v^+) \le 0$ is equivalent to the inequality in (14). Thus, $v^+ = v_A \in \mathcal{S}$, the previous observation implies that $\zeta(v_A) \le 0$. Using that function ζ is continuous on its domain, then there exists $\delta > 0$ such that $\zeta(u) \le 0$ for any $u \in v_A + \mathcal{B}_\delta$, where $\mathcal{B}_\delta$ is the closed ball centered at the origin with radius δ, defined as

$$\mathcal{B}_\delta := \{ z \in \mathbb{R}^n \mid \|z\| \le \delta \}.$$

We now use the facts that $0 \in \mathcal{A} + v_A$ (due to $-v_A \in \mathcal{A}$) and that $\mathcal{A} + v_A$ is convex, to conclude that, if $w \in \mathcal{A} - v_A$ and $t \in [0,1]$, then $tw = tw + (1-t)0 \in \mathcal{A} - v_A$. Hence, $t(\mathcal{A} - v_A) \subset \mathcal{A} - v_A$, for any $t \in [0,1]$. Moreover, since $\mathcal{A} - v_A$ is bounded, there exists $C > 0$ such that $t(\mathcal{A} - v_A) \subset \mathcal{B}_\delta$ for any $t \in [0, C]$. Thus we conclude that

$$C'(\mathcal{A} - v_A) \subset (\mathcal{A} - v_A) \cap \mathcal{B}_\delta,$$

where $C' := \min\{1, C\}$, which is equivalent to

$$\mathcal{R} := C'(\mathcal{A} - v_A) + v_A \subset \mathcal{A} \cap (v_A + \mathcal{B}_\delta).$$

The above set inclusion and the definition of δ imply that, if $u \in \mathcal{R}$, then we have both $u \in \mathcal{A}$ and $\zeta(u) \leq 0$, and thus $u \in \mathcal{S}$. Hence, $\mathcal{R} \subset \mathcal{S}$. As $\mathcal{R}$ is related to $\mathcal{A}$ by a translation and a nonzero scaling factor, it has the same covering dimension as $\mathcal{A}$; from this and from the fact that $\mathcal{R} \subset \mathcal{S} \subset A$, we conclude that the covering dimension of $\mathcal{S}$ is equal to the covering dimension of $\mathcal{A}$.

(b) Since $-v_A = \measuredangle A$ (due to Lemma 2), it follows from (5) that $-v_A$ is the point with the least norm in $\mathcal{A}$. Therefore, since $v^+ \in \mathcal{A}$,

$$\|v^+\|^2 \geq \|v_A\|^2,$$

and thus

$$\psi(A, v^+) \leq -\frac{\theta}{2}\|v^+\|^2 \leq -\frac{\theta}{2}\|v_A\|^2.$$

(c) Replacing $\psi(A, v_A) = -(1/2)\|v_A\|^2$ (derived from Lemma 1) in the inequality in (14), we obtain that

$$\psi(A, v^+) \leq \theta\psi(A, v).$$

The first identity in (13) implies that

$$\psi(A, v^+) \geq \psi(A, v_A).$$

The assumption that $\theta = 1$ and the last two inequalities allow us to conclude that $\psi(A, v^+) = \psi(A, v_A)$. Since the function $\eta : \mathbb{R}^n \to \mathbb{R}$, defined by $\eta(u) = \psi(A, u)$, is strictly convex (and consequently has a unique minimizer), we can conclude that $v^+ = v_A$.

(d) Take a subsequence (v_{n_k}) with $v_{n_k} \to v_A$. Then, for any $\epsilon > 0$, there exists $K \in \mathbb{N}$ such that $\|v_{n_k} - v_A\| < \epsilon$ for all $k \geq K$. Using $\epsilon = \delta$ as in statement (a), we conclude that $\|v_N - v_A\| \leq \delta$ for $N := n_K$. From the definition of (v_k), $v_N \in \mathcal{A}$, and thus $v_N \in \mathcal{S}$.

Remark 1 By Lemma 5, we conclude that it makes sense to take $v^+ \in \mathcal{S}$ as an inexact solution for the problem $\arg\min_u \psi(A, u)$, where $\mathcal{S}$ is defined as in (14). Statement (a) shows the existence and inexactness (for $\theta < 1$) of this choice. Statement (b) shows that this choice is an approximated solution for the problem. Statement (c) shows that this choice degenerates to the exact solution if $\theta = 1$.

Statement (d) shows that, for $\theta < 1$, if an iterative algorithm produces a sequence in $\mathcal{A}$ that has v_A as one of its accumulation points, then such algorithm finds an inexact solution $v^+ \in \mathcal{S}$ in a finite number of iterations. When using such algorithm, we can test if a valid v^+ was found in a given iteration simply by checking if (14) is true for the iteration point, stopping if that is the case. When finding v^+ in this way, if we stop at the first iteration in which (14) is satisfied (which is not necessary, as we can instead run for a

number of iterations and only them test), the larger θ is, the smaller $\psi(A, v^+)$ will be, but the more iterations will be required to find such v^+. As will be discussed later, a quadratic programming solver can be used to efficiently find inexact solution in this way. In some cases it may be more advantageous to use closed-form methods; They can provide an exact or approximate solution in just one step; the comparison of different methods for finding $v^+ \in \mathcal{S}$ is beyond the scope of this work.

3.1 Finite-Difference and Inexactness Bounds

In this subsection, we establish key properties of finite-difference approximations and related quantities that are central to the analysis of the proposed method. We begin by recalling the definition of the forward finite-difference operator and presenting a classical bound on its approximation error with respect to the gradient of a smooth function. This is followed by a sequence of technical lemmas that play an essential role in the subsequent section.

Let $\mathcal{F}_n$ be the set of the real-valued functions of $\mathbb{R}^n$. Let $d_h : \mathcal{F}_n \to \mathbb{R}^{1\times n}$ be the forward finite-difference operator with step $h > 0$, defined by

$$(d_h f(x))_i = \frac{f(x + he_i) - f(x)}{h}, \quad i \in \mathcal{I}_n := \{1, \ldots, n\}, \tag{15}$$

$$(e_i)_j := \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{if } i \neq j \end{cases}. \tag{16}$$

Note that the image of d_h is in $\mathbb{R}^{1\times n}$, so that it is a row vector in calculations.

The next lemma gives an upper bound for the distance between the gradient of a function and its finite-difference approximation. It is a well-known result and is presented in Section 8.1 of [17].

Lemma 6 *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a differentiable function with $\tilde{L}$-Lipschitz continuous gradient and d_h satisfying conditions* (15) *and* (16) *with step $h > 0$. Then,*

$$\|\nabla f(x) - d_h f(x)\| \leq \frac{\sqrt{n}\tilde{L}}{2} h.$$

The next result plays an important role in the algorithm presented in the next section. It establishes that if x, x^- and an approximation for the gradient of F are given, then a key inequality is always guaranteed.

Lemma 7 *Suppose that $\theta \in (0, 1)$ and that assumption A1 holds. Let $x, x^- \in \mathbb{R}^n$ and $G \in \mathbb{R}^{m\times n}$ with*

$$\|JF(x) - G\|_{2,\infty} \leq K\|x - x^-\|, \tag{17}$$

where $K > 0$. Let $\sigma > 0$ be such that

$$x^+ = x + \frac{1}{\sigma} v^+, \tag{18}$$

where $v^+ \in \mathbb{R}^n$ *is such that*

$$\psi(G, v^+) \leq -\frac{\theta}{2}\|\prec G\|^2. \tag{19}$$

If

$$\sigma \geq 2(L + K),$$

then

$$f_i(x) - f_i(x^+) \geq \frac{\sigma}{4}\|x^+ - x\|^2 - \frac{K}{2}\|x - x^-\|^2 \quad \forall i \in \mathcal{I}.$$

Proof Let g_i being the i-th row of G. It follows from (19) and (6) that

$$\max_i g_i v^+ + \frac{1}{2}\|v^+\|^2 \leq -\frac{\theta}{2}\|\prec G\|^2 \leq 0.$$

Hence,

$$g_i v^+ + \frac{1}{2}\|v^+\|^2 \leq 0, \quad \forall i \in \mathcal{I}.$$

Using (18), the last inequality and the assumption that $\sigma > 0$, we obtain that

$$g_i(x^+ - x) + \frac{\sigma}{2}\|x^+ - x\|^2 \leq 0, \quad \forall i \in \mathcal{I}. \tag{20}$$

On the other hand, the definition of $\|\cdot\|_{2,\infty}$ in (4) and (17) imply that

$$\|\nabla f_i(x) - g_i\| \leq K\|x - x^-\|, \quad \forall i \in \mathcal{I}. \tag{21}$$

The previous inequality combined with assumption A1, (20), the Cauchy-Shwartz inequality, (17), (4) and (21), imply that for every $i \in \mathcal{I}$ and $x, x^- \in \mathbb{R}^n$, there holds

$$\begin{aligned}
f_i(x^+) - f_i(x) &\leq \langle g_i, x^+ - x\rangle + \frac{\sigma}{2}\|x^+ - x\|^2 + \langle \nabla f_i(x) - g_i, x^+ - x\rangle + \frac{L-\sigma}{2}\|x^+ - x\|^2 \\
&\overset{(20)}{\leq} \langle \nabla f_i(x) - g_i, x^+ - x\rangle + \frac{L-\sigma}{2}\|x^+ - x\|^2 \\
&\leq \|\nabla f_i(x) - g_i\|\|x^+ - x\| + \frac{L-\sigma}{2}\|x^+ - x\|^2 \\
&\overset{(4),(21)}{\leq} \|JF(x) - G\|_{2,\infty}\|x^+ - x\| + \frac{L-\sigma}{2}\|x^+ - x\|^2 \\
&\overset{(17)}{\leq} K\|x - x^-\|\|x^+ - x\| + \frac{L-\sigma}{2}\|x^+ - x\|^2 \\
&\leq \frac{K}{2}\|x^+ - x^-\|^2 + \frac{K}{2}\|x^+ - x\|^2 + \frac{L-\sigma}{2}\|x^+ - x\|^2 \\
&= \frac{K}{2}\|x^+ - x^-\|^2 - \frac{\sigma - L - K}{2}\|x^+ - x\|^2.
\end{aligned}$$

The conclusion follows from the previous inequality and from the fact that $\sigma \geq 2(L + K)$.

The next lemma gives an upper bound for $\|\measuredangle JF(x^+)\|$ in terms of $(x^+ - x)$ and $(x - x^-)$, when $(x^+ - x)$ is proportional to an inexact solution of $\arg\min_u \psi(G, u)$ and G is an approximation for $JF(x)$ expressed in terms of $(x^- - x)$.

Lemma 8 *Suppose that assumption A1,* (17) *and* (18) *hold. Let x, x^-, G and x^+ be as in Lemma 7. Let $v^+ \in \mathbb{R}^n$ be such that*

$$-v^+ \in \operatorname{conv}\{g_1, \ldots, g_m\}, \tag{22}$$

where g_i is the i-th row of G. Then,

$$\|\measuredangle JF(x^+)\| \leq (\sigma + L + K) \max\{\|x^+ - x\|, \|x - x^-\|\}.$$

Proof It follows from Lemma 4 and the Cauchy-Shwartz inequality for norms that

$$\begin{aligned}\|\measuredangle JF(x^+)\| &\leq \|\measuredangle G\| + \|JF(x^+) - G\|_{2,\infty} \\ &= \|\measuredangle G\| + \|JF(x^+) - JF(x) + JF(x) - G\|_{2,\infty} \\ &\leq \|\measuredangle G\| + \|JF(x^+) - JF(x)\|_{2,\infty} + \|JF(x) - G\|_{2,\infty}. \end{aligned} \tag{23}$$

Using (22), the fact that $\measuredangle G$ is the point with least norm in $\operatorname{conv}\{g_1, \ldots, g_m\}$ and (18), we have

$$\|\measuredangle G\| \leq \|v^+\| = \sigma\|x^+ - x\|. \tag{24}$$

Assumption A1 and the definition of $\|\cdot\|_{2,\infty}$ in (4) imply that

$$\|JF(x^+) - JF(x)\|_{2,\infty} = \max_i\{\|\nabla f_i(x^+) - \nabla f_i(x)\|\} \leq \max_i\{L\|x^+ - x\|\} = L\|x^+ - x\|.$$

Using (17), (24) and the last inequality in (23), we get

$$\begin{aligned}\|\measuredangle JF(x^+)\| &\leq \sigma\|x^+ - x\| + L\|x^+ - x\| + K\|x - x^-\| \\ &\leq (\sigma + L + K) \max\{\|x^+ - x\|, \|x - x^-\|\},\end{aligned}$$

which concludes the proof. □

The following lemma gives an upper bound for a term related to a certain sequence.

Lemma 9 *Given $\tau, \rho \geq 0$ and a set $\{z_j\}_{j=1}^k$ of nonnegative real numbers, with $k \geq 2$, let*

$$m(k) := \operatorname*{arg\,min}_{j \in \{1, \ldots, k-1\}} (z_j^\tau + z_{j+1}^\tau).$$

If $\sum_{j=1}^k z_j^\tau \leq \rho$, then

$$\max\left\{z_{m(k)}, z_{m(k)+1}\right\} \leq \left(\frac{2\rho}{k-1}\right)^{\frac{1}{\tau}}.$$

Proof See Lemma 4 in [9].

4 Proposed forward finite-difference method

In this section, the proposed forward finite-difference method is described as well as its complexity analysis.

Consider the finite-difference matrix operator D_h, defined as

$$D_hF(x) = \begin{bmatrix} d_hf_1(x) \\ \vdots \\ d_hf_m(x) \end{bmatrix} \tag{25}$$

(see (15),(16)). In the proposed method, the update direction is calculated similarly as in the original multiobjective gradient method [6], with the difference that the Jacobian $JF(x_k)$ is approximated with the finite-difference matrix $D_hF(x_k)$. The finite difference step length h and the update length are computed in a manner inspired by the finite-difference algorithm proposed in [7].

Algorithm 1 Finite-difference multiobjective steepest descent

Step 0: $x_0, x_1 \in \mathbb{R}^n$, $x_0 \neq x_1$, $\sigma_1 > 0$, $\theta \in (0, 1]$, $\beta > 0$ and set $k := 1$.
Step 1 If $\sigma_k < 2\sigma_1$, set $j := 1$; otherwise, set $j := 0$.
Step 1.1 Set

$$h_k = \frac{\beta\sigma_1\|x_k - x_{k-1}\|}{\sqrt{n}2^j\sigma_k}.$$

Step 1.2 Find v^+ such that

$$-v^+ \in \text{conv}\{d_hf_1(x_k), \ldots, d_hf_m(x_k)\} \tag{26}$$

$$\psi(D_hF(x_k), v^+) \leq -\frac{\theta}{2}\|v^+\|^2 \tag{27}$$

and set

$$x^+_{k,j} = x_k + \frac{1}{2^j\sigma_k}v^+.$$

Step 1.3 If

$$f_i(x_k) - f_i(x^+_{k,j}) \geq \frac{2^j\sigma_k}{4}\|x^+_{k,j} - x_k\|^2 - \frac{\sigma_1}{4}\|x_k - x_{k-1}\|^2, \quad \forall i \in \mathcal{I}, \tag{28}$$

set $j_k = j$ and go to **Step 2**. Otherwise, set $j := j + 1$ and go to **Step 1.1**.
Step 2 Set $x_{k+1} = x^+_{k,j}$, $\sigma_{k+1} = 2^{j_k-1}\sigma_k$, $k := k + 1$ and go to **Step 1**.

Remark 2 It follows from Lemma 5 (with $\theta = 1$) that the only v^+ satisfying (27) is

$$v^+ = \arg\min_v \psi(D_hF(x_k), v) = -\nsucc D_hF(x_k),$$

where $D_hF(\cdot)$ is as in (25). Thus, by Lemma 2, the calculation of v^+ in Step 1.2 of Algorithm 1 can be made by solving the problem

$$\lambda = \arg\min_{y \in \Delta_m} \|(D_hF(x_k))^T y\|, \tag{29}$$

and setting $v^+ = (D_h F(x_k))^T \lambda$. As discussed before, this can be made with a QP solver. In practice, however, a QP solver only gets an approximated solution, and thus such approach is not implementable for $\theta = 1$. If the subproblem is solved using an exact, non-iterative method (such as the recursive approach proposed in [10]) rather than a QP solver, the resulting approximate solution may be very close to the true one. However, due to unavoidable numerical errors, some level of inexactness is still expected, which results in the effective value of θ being strictly less than 1.

For $\theta < 1$, an inexact minimization of $\psi(D_h F(x_k), u)$ in u is sufficient to obtain a v^+ satisfying (26) and (27). Indeed, by Lemma 5(d), any iterative method that produces a sequence that has the exact solution $-\measuredangle D_h F(x_k)$ as an accumulation point finds such v^+ in a finite number of iterations. Thus, we can proceed as follows. Use some algorithm to get an approximated solution $\lambda' \in \Delta_m$ for (29); set $v^+ = (D_h F(x))^T \lambda'$. Such approximation λ' can be obtained with a finite number of iterations of QP. Given an approximated solution v^+ obtained this way, the condition (26) will be satisfied automatically by the fact that $\lambda \in \Delta_m$. The condition (27) can be directly verified, since all terms are known. If the condition (27) is not satisfied, then the algorithm that calculated the approximated solution λ' for (29) continues to get more refined approximation λ'', and v^+ is updated with $v^+ = (D_h F(x))^T \lambda''$. If still v^+ do not satisfy the condition (27), then this procedure continues. By Lemma 5.5, this procedure is guaranteed to produce a valid v^+ in a finite number of iterations. This way, for $\theta < 1$, v^+ satisfying the conditions is guaranteed to be obtained with a finite number of iterations of QP.

In practice, if the QP solver used for approximating λ is set to a typically used tolerance, it is plausible to expect that there will be some $\theta \in (0, 1)$ such that (27) is always satisfied for this given tolerance. Therefore, we can expect the Algorithm 1 to work without testing (27). If we want to ensure that (27) is satisfied for a manually set value of θ, we use the procedure described in the previous paragraph.

The following lemma shows that, under Assumption A1, the sequence $\{\sigma_k\}$ generated by Algorithm 1 remains uniformly bounded above and below. This property will be useful in the convergence and complexity analysis.

Lemma 10 *Suppose that A1 holds. Then, the sequence of inverse step lenghts σ_k in Algorithm 1 satisfies*

$$\sigma_1 \leq \sigma_k \leq 2\left(L + \frac{\sigma_1}{2}\right) =: \sigma_{\max},$$

for all $k \geq 1$.

Proof To prove this, we can proceed exactly as in [7].

The following theorem provides a worst-case complexity bound for Algorithm 1. Under standard assumptions, it guarantees that the number of iterations required to reduce the optimality measure $\|\measuredangle JF(x_k)\|$ below a given threshold $\epsilon > 0$ grows at most like $\mathcal{O}(\epsilon^{-2})$.

Theorem 1 *Suppose that assumptions A1 and A2 hold and let $(x_k)_{k\geq 0}$ be the sequence generated by Algorithm 1. Suppose that for any given tolerance $\epsilon > 0$ and $T \in \mathbb{N}$, there holds*

$$\|\measuredangle JF(x_k)\| > \epsilon, \quad k = 1, \ldots, T. \tag{30}$$

Then,

$$T \leq 3 + 2RS^2\epsilon^{-2}, \tag{31}$$

where

$$S := \sigma + L + \frac{\beta L}{2}, \quad R := \frac{4\big(\max_i\{f_i(x_1) - l_i\}\big)}{\sigma_1} + \|x_1 - x_0\|^2. \tag{32}$$

Proof It follows from assumption A2 that there exists $i' \in \mathcal{I}$ such that

$$f_{i'}(x_1) - l_{i'} \geq f_{i'}(x_1) - f_{i'}(x_T).$$

As a consequence,

$$\max_i\{f_i(x_1) - l_i\} \geq f_{i'}(x_1) - l_{i'} \geq f_{i'}(x_1) - f_{i'}(x_T). \tag{33}$$

Since $2^{j_k}\sigma_k = 2\sigma_{k+1}$ and $x_{k+1} = x_{k,j}^+$ (due to Step 2 of Algorithm 1), and $\sigma_k \geq \sigma_1$ for every $k \geq 1$ (due to Lemma 10), it follows from the acceptance condition (28) that for every $i \in \mathcal{I}$, there holds

$$f_i(x_k) - f_i(x_{k+1}) \geq \frac{\sigma_{k+1}}{2}\|x_{k+1} - x_k\|^2 - \frac{\sigma_k}{4}\|x_k - x_{k-1}\|^2, \quad k = 1, \ldots, T-1. \tag{34}$$

Let $i^* \in \arg\min_i \{f_i(x_1) - f_i(x_T)\}$. Inequalities (33) and (34), together with the fact that $\sigma_k \geq \sigma_1$, imply that

$$\begin{aligned}
\max_i\{f_i(x_1) - l_i\} \geq f_{i^*}(x_1) - f_{i^*}(x_T) &= \sum_{k=1}^{T-1}\big(f_{i^*}(x_k) - f_{i^*}(x_{k+1})\big) \\
&\overset{(34)}{\geq} \sum_{k=1}^{T-1}\Big(\frac{\sigma_{k+1}}{2}\|x_{k+1} - x_k\|^2 - \frac{\sigma_k}{4}\|x_k - x_{k-1}\|^2\Big) \\
&= \sum_{k=1}^{T-1}\frac{\sigma_{k+1}}{2}\|x_{k+1} - x_k\|^2 - \sum_{k=1}^{T-1}\frac{\sigma_k}{4}\|x_k - x_{k-1}\|^2 \\
&= \sum_{k=2}^{T}\frac{\sigma_k}{2}\|x_k - x_{k-1}\|^2 - \sum_{k=2}^{T-1}\frac{\sigma_k}{4}\|x_k - x_{k-1}\|^2 - \frac{\sigma_1}{4}\|x_1 - x_0\|^2 \\
&\geq \sum_{k=2}^{T}\frac{\sigma_k}{2}\|x_k - x_{k-1}\|^2 - \sum_{k=2}^{T}\frac{\sigma_k}{4}\|x_k - x_{k-1}\|^2 - \frac{\sigma_1}{4}\|x_1 - x_0\|^2 \\
&= \sum_{k=2}^{T}\frac{\sigma_k}{4}\|x_k - x_{k-1}\|^2 - \frac{\sigma_1}{4}\|x_1 - x_0\|^2 \\
&\geq \frac{\sigma_1}{4}\sum_{k=1}^{T-1}\|x_{k+1} - x_k\|^2 - \frac{\sigma_1}{4}\|x_1 - x_0\|^2.
\end{aligned}$$

Therefore,

$$\sum_{k=1}^{T-1} \|s_k\|^2 \leq \frac{4\big(\max_i\{f_i(x_1) - l_i\}\big)}{\sigma_1} + \|s_0\|^2 = R,$$

where $s_k := x_{k+1} - x_k$, for $k = 1, \ldots, T-1$, and R is as in (32).

Observe that if $T < 3$, (31) is automatically satisfied. Thus, assume that $T \geq 3$ and let

$$t = \underset{j \in \{1,\ldots,T-2\}}{\arg\min} \big(\|s_j\|^2 + \|s_{j-1}\|^2\big).$$

Then, by Lemma 9 with $z_j = \|s_j\|$, $k = T-1$, $\tau = 2$ and $\rho = R$, it follows that

$$\max\{\|s_t\|, \|s_{t+1}\|\} \leq \left(\frac{2R}{T-2}\right)^{\frac{1}{2}}. \tag{35}$$

The condition in Step 1.1, Lemma 6, and the fact that $\sigma_{t+1} \geq \sigma_1$ (due to Lemma 10), imply that

$$\|\nabla f_i(x_{t+1}) - d_{h_k} f_i(x_{t+1})\| \leq \frac{\sqrt{n}L}{2} \frac{\beta\sigma_1 \|x_{t+1} - x_t\|}{\sqrt{n} 2^{j_{t+1}} \sigma_{t+1}} \leq \frac{\beta L}{2} \|x_{t+1} - x_t\|.$$

By (30), Lemma 8 with $K = \beta L/2$ and $G = D_{h_{t+1}} F(x_{t+1})$, and by (35),

$$\begin{aligned}
\epsilon &< \|\nsucc JF(x_{t+2})\| \\
&\leq \left(\sigma + L + \frac{\beta L}{2}\right) \max\{\|x_{t+2} - x_{t+1}\|, \|x_{t+1} - x_t\|\} \\
&= S \max\{\|s_t\|, \|s_{t+1}\|\} \\
&\leq S \left(\frac{2R}{T-2}\right)^{\frac{1}{2}}.
\end{aligned}$$

Therefore, $T \leq 3 + 2RS^2\epsilon^{-2}$. □

The next result gives an upper bound for the total number of function evaluations that Algorithm 1 performs until it finds an ϵ-approximate stationary point of the objective function. Its proof is similar to the proof of Corollary 1 in [7].

Corollary 1 *Assume that assumptions A1 and A2 hold. Let $\mathcal{I}(\epsilon)$ be the first iteration index such that $\|\nsucc JF(x_{\mathcal{I}(\epsilon)})\| \leq \epsilon$ and $\mathcal{FE}(\epsilon)$ be the total number of function evaluations performed by Algorithm 1 up to the $(\mathcal{I}(\epsilon)-1)$-th iteration. Then,*

$$\mathcal{FE}(\epsilon) \leq 1 + m(n+1)[2\mathcal{I}(\epsilon) + \log_2(\sigma_{\max}) - \log_2(\sigma_0)],$$

where $\sigma_{\max}$ is defined as in Lemma 10. As a consequence $\mathcal{FE}(\epsilon) = \mathcal{O}(mn\epsilon^{-2})$.

Proof We first observe that the number of function evaluations performed at the k-th iteration of Algorithm 1 is bounded from above by $1+m(n+1)(j_k+1)$. On the other hand, the fact that $\sigma_{k+1} = 2^{j_k-1}\sigma_k$ implies that

$$m(n+1)(j_k+1) = m(n+1)[2 + \log_2(\sigma_{k+1}) - \log_2(\sigma_k)].$$

As a consequence, we have

$$\begin{aligned}\mathcal{FE}(\epsilon) &\leq 1 + m(n+1)\sum_{k=0}^{\mathcal{I}(\epsilon)-1}(j_k+1)\\ &= 1 + m(n+1)[2\mathcal{I}(\epsilon) + \log_2(\sigma_{\mathcal{I}(\epsilon)}) - \log_2(\sigma_0)]\\ &\leq 1 + m(n+1)[2\mathcal{I}(\epsilon) + \log_2(\sigma_{\max}) - \log_2(\sigma_0)],\end{aligned}$$

where the last inequality is due to Lemma 10. The conclusion of the last part of this result follows immediately from the first part combined with (31). □

5 Numerical experiments

In this section, we investigate the practical performance of the proposed finite-difference steepest descent method (referred to as FD-SD in this section) by comparing it to the performance of the original multiobjective steepest descent method (referred to as SD in this section) in finding Pareto-critical points (PC) points of the test function $F : \mathbb{R}^2 \to \mathbb{R}^2$ defined by

$$F(x) := \begin{bmatrix}\cos(a(x))b(x)\\ \sin(a(x))b(x)\end{bmatrix},$$

with

$$\begin{aligned}a(x) &:= a_c + a_1\sin(x_1) + a_2\sin(x_2)\\ b(x) &:= 1 + d\cos(x_1).\end{aligned}$$

This function is equivalent to the one defined in Example 4.1 of [11], except for scaling changes. The choices for the coefficients were

$$a_c = 0.785, \quad a_1 = 0.698, \quad a_2 = 0.436, \quad d = 0.5.$$

It is possible to show that, if the coefficients satisfy a few simple relations (which are out of the scope of this document), then the set of Pareto-critical points are the lines

$$x = k\pi,\ k \in \mathbb{Z}; \quad y = \left(\frac{1}{2} + k\right)\pi,\ k \in \mathbb{Z}.$$

Such relations are satisfied in special by the combination of coefficients adopted. Figure 1 shows the functional values for a grid of lines and for the Pareto-critical points.

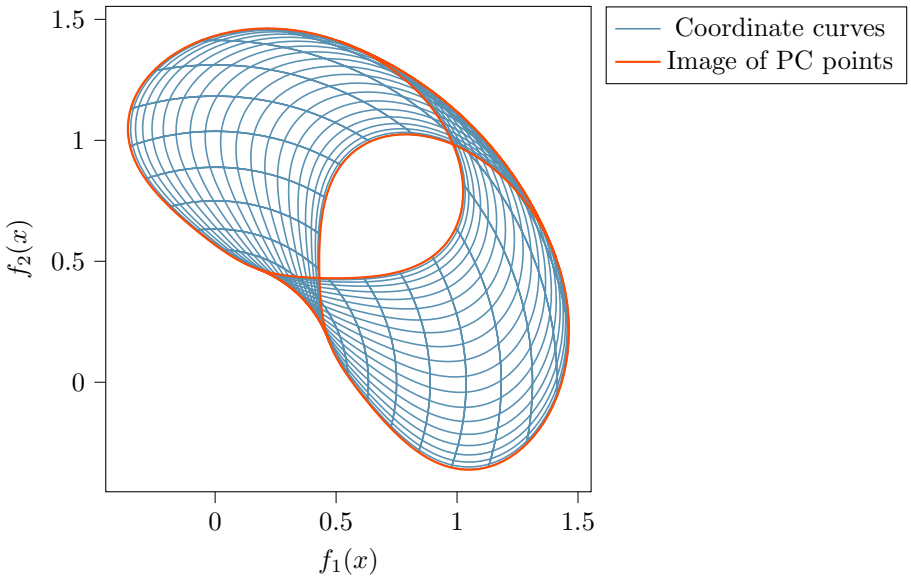

Fig. 1 Functional values of the test function used in the experiment.

A total of 100 points were generated to be used as starting points. The same set of points was used for both algorithms and in all parts of the experiment. The test function is periodic with period 2π in both x_1 and x_2; therefore, to cover all regions of the functions, except those equivalent due to the periodicity, both coordinates of the initial points were generated using the uniform distribution with support $[0, 2\pi]$.

The parameter choices for FD-SD were

$$\|x_0 - x_1\| = \sigma_1 = 0.1$$

throughout the experiment.

The first part of the experiment determined the number of iterations required for SD and FD-SD to find an ϵ-stationary point, which here means finding a point x such that $\|\measuredangle JF(x)\| \leq \epsilon$, where the values chosen for ϵ were 10^{-3} and 10^{-6}. In this part of the experiment, the ϵ-stationary condition had to be tested at each iteration for both algorithms. Hence, the exact steepest decent direction $\measuredangle JF(x)$ had to be calculated even for FD-SD. It is worth to mentioned, that the vector $\measuredangle JF(x)$ was used in the FD-SD routine just in order to determine the number of iterations taken to reach the selected tolerances for each initial point; the result of $\measuredangle JF(x)$ was *not* used by FD-SD in any calculation except to check if the tolerances were reached, and the calculation of $\measuredangle JF(x)$ is not required in FD-SD.

The second part of the experiment measured the time interval taken to run the amount of iterations obtained in the first part. For each initial point, the time elapsed until each algorithm reached the required number of iterations for

each tolerance was measured 10 times. Since the required number of iterations for each combination of initial point, algorithm and tolerance was already obtained in the first part, in this part there was no testing inside the algorithm routines to check if the ϵ-stationary condition was met. Thus, in this part FD-SD routine did not compute $\measuredangle JF(x)$, allowing a fair comparison between the time performance of SD and FD-SD.

Note that, if the time intervals were measured in the first part instead, then the comparison would be unfair, as in the first part there was the calculation of $\measuredangle JF(x)$ in each iteration of the FD-SD routine, which increased the computational cost of the routine. This was the reason for first determining the required iterations and then measuring the time intervals.

The figures 2 and 3 show the sequences produced by each algorithm for the first 35 points and their images (functional values), respectively, for tolerance 10^{-6}. Note that, for a same initial point, the trajectories produced by both algorithms may be quite different. This is the expected behavior; for a same point, each algorithm produces a update that has different direction and length; at the first iteration, the finite difference step h_k is not small, and a large difference between the update directions is expected; thus, the points of the algorithms for the next iteration are not expected to be close; in the subsequent iterations, h_k reduces, but as the second points of the trajectories are not necessarily close, the remaining parts of the trajectories are expected to be very different. This behavior makes it essential to use multiple iteration points to compare the performance of the algorithms reliably.

The table 1 shows the mean and standard deviation of the required number of iterations and the time interval for each tolerance and algorithm. The mean and standard deviation are calculated with respect to the 100 initial points for the number of iterations and with respect to the 1000 trials (100 initial points times 10 trials for each initial point) for the time intervals. The results show that, for this specific experiment, on average, the proposed FD-SD takes about 50% and 40% as many iterations as SD to reach the tolerances 10^{-3} and 10^{-6}, respectively. On the other hand, on average, the CPU times were similar for both algorithms, with FD-SD requiring more time than SD for tolerance 10^{-3} and the opposite occurring for tolerance 10^{-6}. The fact that the CPU times were not approximately proportional to the iteration times is explained by the fact that, in a single iteration of FD-SD, Steps 1.1 to 1.3 are repeated until the acceptance condition is met, essentially multiplying the computational cost of some iterations. This indicates that a large gain in performance can be achieved with modifications that avoid the recalculation of h_k in a given iteration, which would also avoid the recalculation of v^+. There is a large variability in the averaged results, indicated by the high standard deviations; this occurs because the number of iterations required by an algorithm to reach a given tolerance depends highly on the initial point.

It was further observed that, for a few initial points, SD required much more iterations to reach the tolerances than for most initial points. This indicates a slow convergence of SD when the limit point is in specific areas of the set

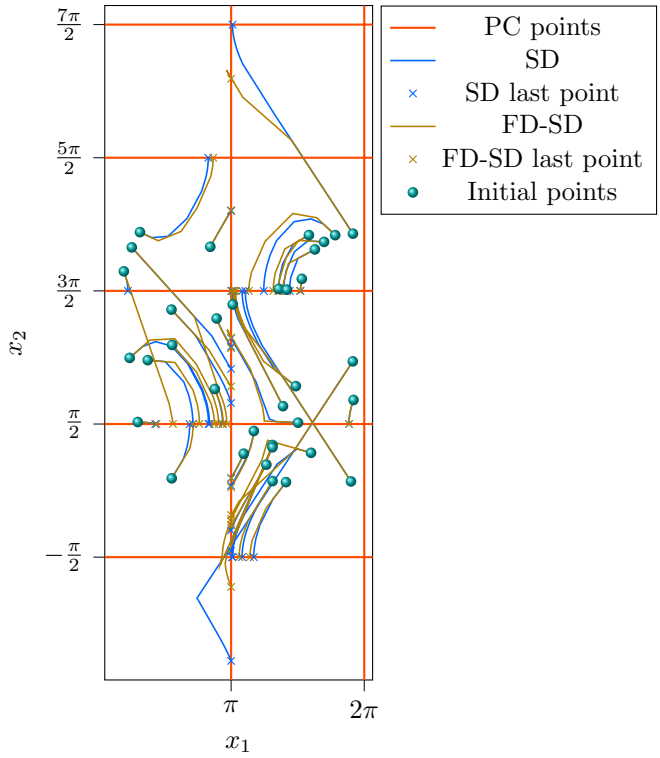

Fig. 2 Sequences

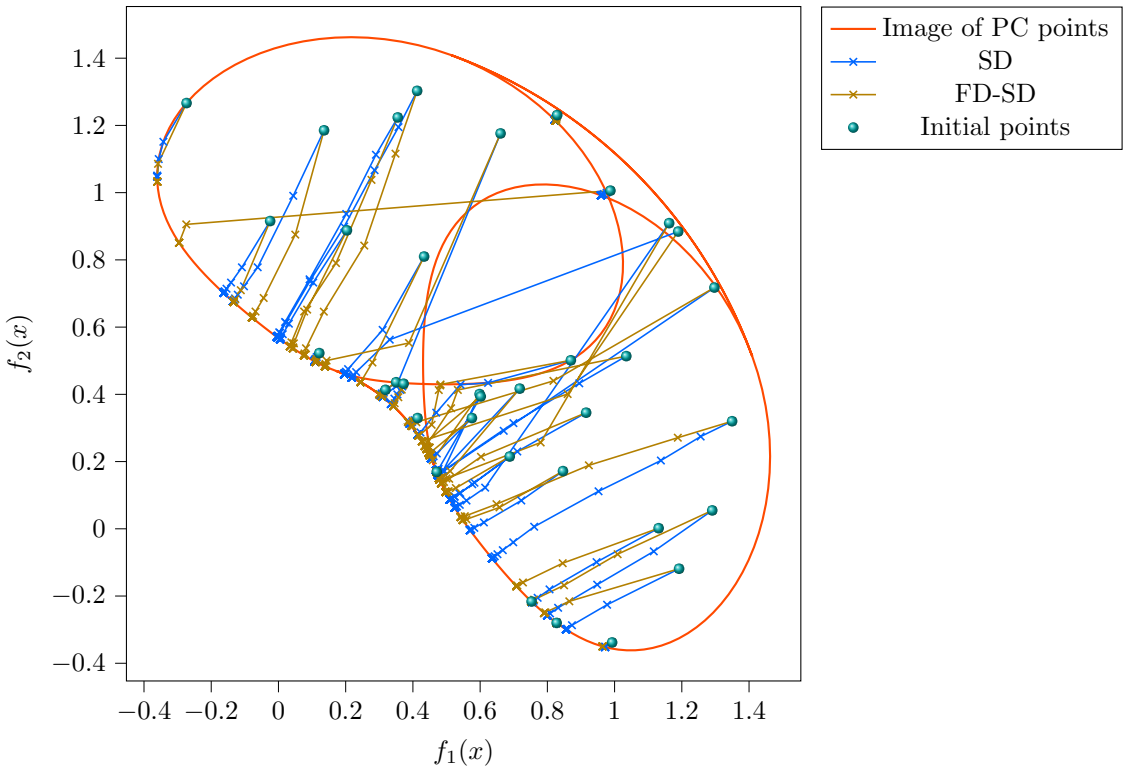

Fig. 3 Image (functional values) of the sequences.

Table 1 Mean and standard deviation of the required number of iterations and the CPU time

Tolerance	Algorithm	Num. iterations		CPU time	
		Mean	Std	Mean	Std
10^{-3}	SD	14.62	10.05	0.01020	0.00814
	FD-SD	8.17	6.00	0.01267	0.00827
10^{-6}	SD	99.15	212.78	0.06572	0.13754
	FD-SD	40.20	96.02	0.05535	0.12453

of Pareto-critical points. Further investigation will be conducted to determine the cause of this behavior. This behavior was not observed for FD-SD.

6 Conclusion

We proposed a quadratic regularization approach employing finite-difference gradient approximations for solving unconstrained multiobjective optimization problems. The algorithm proposed features an adaptive finite-difference step and a backtracking strategy inspired by recent developments in single-objective optimization. A key advantage of the method is that it does not require explicit computation of gradients, making it particularly suitable for problems where only function values are available. We have shown that the inverse step sizes remain uniformly bounded and provided a worst-case complexity bound for the number of iterations needed to achieve a desired level of optimality. Specifically, the iteration complexity is of order $\mathcal{O}(\epsilon^{-2})$, matching the known bounds for first-order methods in nonconvex multiobjective optimization. These results establish the method as a viable and theoretically grounded approach for derivative-free multiobjective optimization.

Future work includes extending the proposed finite-difference method to constrained multiobjective problems, particularly those with equality and inequality constraints. Incorporating proximal or augmented Lagrangian techniques may help preserve convergence guarantees in such settings. Second, it is of interest to adapt the method for composite optimization problems of the form $F + G$, where $F : \mathbb{R}^n \to \mathbb{R}^m$ is a vector-valued function, and $G : \mathbb{R}^n \to \mathbb{R}^m \cup \{+\infty\}$ is a proper convex lower semicontinuous function. Developing proximal-type extensions of Algorithm 1 that can handle such composite terms without requiring exact gradients would significantly broaden its applicability, especially in high-dimensional and structured settings.

References

1. Charles Audet and Dominique Orban. Finding optimal algorithmic parameters using derivative-free optimization. *SIAM Journal on Optimization*, 17(3):642–664, 2006.
2. Stefania Bellavia, Gianmarco Gurioli, Benedetta Morini, and Philippe L Toint. Quadratic and cubic regularisation methods with inexact function and random deriva-

tives for finite-sum minimisation. In *2021 21st International Conference on Computational Science and Its Applications (ICCSA)*, pages 258–267. IEEE, 2021.
3. L Calderón, MA Diniz-Ehrhardt, and José Mario Martínez. On high-order model regularization for multiobjective optimization. *Optimization Methods and Software*, 37(1):175–191, 2022.
4. Coralia Cartis, Nicholas IM Gould, and Ph L Toint. On the complexity of steepest descent, newton's and regularized Newton's methods for nonconvex unconstrained optimization problems. *SIAM Jurnal on Optimization*, 20(6):2833–2852, 2010.
5. LM Grana Drummond and Benar F Svaiter. A steepest descent method for vector optimization. *Journal of computational and applied mathematics*, 175(2):395–414, 2005.
6. Jörg Fliege and Benar Fux Svaiter. Steepest descent methods for multicriteria optimization. *Mathematical methods of operations research*, 51:479–494, 2000.
7. Geovani Nunes Grapiglia. Quadratic regularization methods with finite-difference gradient approximations. *Computational Optimization and Applications*, 85(3):683–703, 2023.
8. Geovani Nunes Grapiglia. Worst-case evaluation complexity of a derivative-free quadratic regularization method. *Optimization Letters*, pages 195—-213, 2024.
9. Geovani Nunes Grapiglia, Max LN Gonçalves, and GN Silva. A cubic regularization of newton's method with finite difference hessian approximations. *Numerical Algorithms*, pages 1–24, 2022.
10. Rob Heylen, Dževdet Burazerovic, and Paul Scheunders. Fully constrained least squares spectral unmixing by simplex projection. *IEEE Transactions on Geoscience and Remote Sensing*, 49(11):4112–4122, 2011.
11. Claus Hillermeier. Generalized homotopy approach to multiobjective optimization. *Journal of Optimization Theory and Applications*, 110(3):557–583, 2001.
12. Hamid R Karbasian and Brian C Vermeire. Gradient-free aerodynamic shape optimization using large eddy simulation. *Computers & Fluids*, 232:105185, 2022.
13. Jakub Konečný and Peter Richtárik. Simple complexity analysis of simplified direct search, 2014.
14. Jeffrey Larson, Matt Menickelly, and Stefan M. Wild. Derivative-free optimization methods. *Acta Numerica*, 28:287–404, May 2019.
15. Qiuwei Li, Daniel McKenzie, and Wotao Yin. From the simplex to the sphere: faster constrained optimization using the hadamard parametrization. *Information and Inference: A Journal of the IMA*, 12(3):1898–1937, 2023.
16. Alison L Marsden, Jeffrey A Feinstein, and Charles A Taylor. A computational framework for derivative-free optimization of cardiovascular geometries. *Computer methods in applied mechanics and engineering*, 197(21-24):1890–1905, 2008.
17. Jorge Nocedal and Stephen J Wright. *Numerical optimization, 2nd edn.* Springer, 2006.
18. Sophy Oliver, Coralia Cartis, Iris Kriest, Simon F B Tett, and Samar Khatiwala. A derivative-free optimisation method for global ocean biogeochemical models. *Geoscientific Model Development*, 15(9):3537–3554, 2022.
19. Jonathan B Russ, Richard L Li, Abigail R Herschman, Haim Waisman, Vijay Vedula, Jeffrey W Kysar, and David Kalfa. Design optimization of a cardiovascular stent with application to a balloon expandable prosthetic heart valve. *Materials & design*, 209:109977, 2021.
20. Benar Fux Svaiter. Holder continuity of the steepest descent direction for multiobjective optimization. *arXiv preprint arXiv:1802.01402*, 2018.